

## Perancangan Sistem Rekomendasi Adaptif Latihan dan Nutrisi Berbasis Reinforcement Learning

Angga Putra Laziman Sadiarta<sup>1\*</sup>, Rifai Alinza Putra<sup>2</sup>, Satria Baihaqi Yaasiin<sup>3</sup>, Syifa Nur Rakhmah<sup>4</sup>, Findi Ayu Sariashih<sup>5</sup>

<sup>1,2,3,4,5</sup>Fakultas Teknik dan Informatika, Program Studi Informatika, Universitas Bina Sarana Informatika

\*15230257@bsi.ac.id

### Abstrak

Penelitian ini bertujuan merancang dan mengembangkan Sistem Rekomendasi Latihan dan Nutrisi Adaptif yang dipersonalisasi, menggunakan pendekatan *Reinforcement Learning (RL)*, untuk mengatasi keterbatasan program kebugaran konvensional yang bersifat generik dan statis. Tren gaya hidup tidak sehat telah meningkatkan penyakit tidak menular, namun solusi kebugaran yang tersedia gagal beradaptasi secara *real-time* terhadap kondisi dinamis pengguna—seperti tingkat kelelahan, perubahan berat badan, atau respons nutrisi—sehingga menjadi faktor utama kegagalan konsistensi. Penelitian ini mengadopsi metodologi *Design Science Research Methodology (DSRM)*. Masalah utama diselesaikan melalui perancangan arsitektur sistem cerdas yang mampu mengelola dan memproses data dinamis pengguna. Algoritma *Deep Q-Network (DQN)* diimplementasikan secara efektif untuk menghasilkan rekomendasi program latihan dan nutrisi yang optimal dan personal secara berkelanjutan. Sistem ini dilatih dan dievaluasi menggunakan data sintetis (*synthetic data*) untuk memungkinkan simulasi skenario *feedback* yang kompleks dan mengatasi isu privasi data sensitif. Keluaran penelitian adalah prototipe fungsional (*Proof-of-Concept*) yang akan dianalisis dan dievaluasi kinerjanya secara komparatif dengan program kebugaran konvensional untuk mengukur tingkat optimalisasi program dan konsistensi pengguna.

Kata kunci : Adaptif, Deep Q-Network, Reinforcement Learning, Sistem Rekomendasi, Sequential Decision-Making

### Abstract

*This research aims to design and develop a personalized Adaptive Exercise and Nutrition Recommendation System, using a Reinforcement Learning (RL) approach, to overcome the limitations of conventional fitness programs that are generic and static. Unhealthy lifestyle trends have increased non-communicable diseases, but available wellness solutions fail to adapt in real-time to users' dynamic conditions—such as fatigue levels, weight changes, or nutritional responses—making them a major factor in consistent failure. This research adopts the Design Science Research Methodology (DSRM) methodology. The main problem is solved through designing an intelligent system architecture that is capable of managing and processing dynamic user data. The Deep Q-Network (DQN) algorithm is implemented effectively to produce optimal and personalized training and nutrition program recommendations on an ongoing basis. The system is trained and evaluated using synthetic data to enable simulation of complex feedback scenarios and address sensitive data privacy issues. The research output is a functional prototype (Proof-of-Concept) which will be analyzed and evaluated for its performance comparatively with conventional fitness programs to measure the level of program optimization and user consistency.*

Keywords : *Adaptif, Deep Q-Network, Reinforcement Learning, Sistem Rekomendasi, Sequential Decision-Making.*

### 1. Pendahuluan

Tren gaya hidup yang tidak sehat dan pola makan yang tidak seimbang telah menimbulkan kekhawatiran global, ditandai dengan

peningkatan signifikan dalam angka penyakit tidak menular seperti obesitas, diabetes tipe 2, dan penyakit kardiovaskular pada usia produktif. Data ini didukung oleh World Health Organization

(WHO) yang menyatakan bahwa CVD sebagai penyebab nomor satu kematian di dunia dan terburuk pada negara-negara berkembang [1]. Ironisnya, meskipun akses terhadap informasi kebugaran dan diet saat ini sangatlah melimpah, banyak individu justru mengalami kebingungan dan kesulitan untuk mempertahankan konsistensi karena program yang tersedia seringkali bersifat generik dan statis. Program diet dan latihan yang tersedia di internet atau bahkan yang direkomendasikan secara konvensional sering kali tidak mempertimbangkan variabilitas unik dari setiap individu, seperti tingkat kebugaran awal, preferensi makanan, atau jadwal harian yang padat.

Aplikasi *fitness* konvensional, yang umumnya berfungsi sebagai buku harian digital atau penyedia *template* latihan, gagal beradaptasi secara mendalam terhadap kondisi dinamis pengguna seperti tingkat kelelahan, perubahan berat badan, atau respons tubuh terhadap jenis nutrisi tertentu. Sebagai contoh, program latihan hari Senin yang dirancang untuk minggu pertama akan sama persis dengan minggu kesepuluh, padahal kemampuan dan kondisi fisik pengguna sudah pasti mengalami perubahan. Ketidakmampuan untuk menyesuaikan diri secara *real-time* ini sering menjadi faktor utama kegagalan dalam mencapai dan mempertahankan target kebugaran.

Kurangnya adaptabilitas ini juga menciptakan risiko bagi pengguna, di mana program latihan yang terlalu intensif tanpa penyesuaian dapat menyebabkan cedera atau kelelahan berlebih (*burnout*), yang pada akhirnya membuat mereka berhenti total dari program kebugaran. Oleh karena itu, solusi yang tersedia saat ini belum mampu memfasilitasi proses pengambilan keputusan berurutan (*sequential decision-making*) yang esensial dalam perjalanan kebugaran jangka panjang.

Keterbatasan solusi ini menunjukkan adanya kebutuhan mendesak akan sebuah sistem cerdas yang mampu membuat keputusan berurutan dan mengoptimalkan program secara *real-time*. Di sinilah paradigma Kecerdasan Buatan (AI), khususnya Reinforcement Learning (RL), hadir sebagai solusi. RL adalah cabang dari machine learning yang berfokus pada pembelajaran tindakan yang optimal melalui percobaan dan pengalaman. RL adalah pendekatan Machine Learning (ML) tradisional yang paling banyak digunakan untuk kontrol sinyal lalu lintas karena tidak memerlukan model pasti dari perilaku arus lalu lintas stokastik dalam jaringan persimpangan [2].

Berbeda dengan model prediksi biasa, RL dirancang untuk belajar dari umpan balik (*feedback*) dan pengalaman pengguna secara berkelanjutan, mirip dengan cara manusia belajar dari coba-coba dalam lingkungan nyata.

Penerapan RL memungkinkan perancangan sistem rekomendasi latihan dan nutrisi yang benar-benar adaptif dan terpusat pada individu (*personalized*). Sistem ini dapat secara otomatis menyesuaikan intensitas latihan (misalnya, menambah beban atau mengurangi repetisi) atau mengubah komposisi nutrisi (misalnya, menambah asupan protein setelah latihan berat) berdasarkan data kinerja dan kondisi terkini pengguna, sehingga memaksimalkan efektivitas program dan meminimalisir risiko cedera. Pendekatan ini diharapkan dapat menjadi terobosan untuk mencapai target kebugaran optimal dengan tingkat konsistensi yang tinggi

## 2. Tinjauan Pustaka

### 2.1. Penelitian Terkait

Dalam membuat penelitian ini, peneliti memiliki bahan referensi serta bahan perbandingan dari penelitian terdahulu, antara lain :

- Walse et al. Melakukan penelitian menggunakan metode mixed yang menggabungkan data historis aktivitas mahasiswa dengan model *Reinforcement Learning* (RL). Fokus utama dari penelitian yang ditulis oleh Waisen dan tim adalah Perancangan dan Implementasi Sistem Rekomendasi Adaptif untuk Kurikulum Pembelajaran Personal pada Platform Edukasi Daring<sup>[3]</sup>

- Arrie Kurniawardhani, S.Si., M.Kom dan Muhammad Rizqi Az Zayyad [4]melakukan

penelian tinjauan literatur berdasarkan implementasi metode Deep Learning di sistem rekomendasi film. Sistem rekomendasi sudah biasa diimplementasikan di berbagai industri, termasuk di dalamnya layanan televisi berbayar, sehingga mencapai produktivitas, efisiensi, dan memberikan layanan terbaik bagi konsumennya. Implementasi sistem rekomendasi Deep Learning adalah metode lebih efisien dan efektif.

- Ardan Dwi Hartono dan Joko Sutopo melakukan penelitian Dalam mempelajari pengembangan sistem rekomendasi musik menggunakan pendekatan Reinforcement Learning untuk mengatasi masalah keseimbangan antara eksplorasi dan eksplorasi<sup>[5]</sup>. Algoritma UCB didasarkan pada prinsip optimisme dalam menghadapi ketidakpastian. Dalam konteks masalah multi-armed bandit, UCB memilih item atau lagu yang akan dipilih berdasarkan rata-rata tawaran dan ketidakpastian terkait itu. UCB menyeimbangkan eksplorasi lagu baru dengan eksplorasi lagu yang telah sering kita dengar dan kami sukai.

- Muhammad Sobri et al melakukan penelitian menggunakan metode *Systematic Literature Review* (SLR) untuk mengusulkan model konseptual yang disebut **uCALM** (*Usability of Cardiovascular Health Monitoring Application Model*)<sup>[1]</sup>. Penelitian ini didorong oleh tingginya angka kematian akibat *Cardiovascular Disease* (CVD) dan perlunya aplikasi *mobile health* yang

dapat digunakan secara efektif oleh pasien dan dokter spesialis jantung

## 2.2. Landasan Teori

### 1. Sistem Rekomendasi

Sistem rekomendasi didefinisikan sebagai sistem yang mengolah informasi pengguna dan memberikan rekomendasi sesuai dengan karakteristik pengguna tersebut, sehingga mempermudah pengguna dalam mengambil keputusan yang tepat [6]. menyatakan dalam jurnal mereka bahwa sistem rekomendasi digunakan terutama untuk mengatasi masalah pencarian informasi yang relevan dari banyaknya kumpulan informasi yang tersedia, dengan tujuan memberikan rekomendasi yang sesuai dengan kebutuhan pengguna. Sistem ini sering diterapkan di berbagai bidang dengan jumlah data yang besar dan beragam.

### 2. Konsep Inti Kecerdasan Buatan

Reinforcement learning adalah bagian dari *artificial intelligence* yang melatih algoritma dengan sistem *trial and error*. RL berinteraksi dengan lingkungannya dan mengamati konsekuensi atas tindakannya sebagai tanggapan atas penghargaan maupun hukuman yang diterima. Informasi yang dihasilkan dari setiap interaksi dengan lingkungan, digunakan RL untuk memperbarui pengetahuannya [15]. Reinforcement Learning (RL) merupakan salah

satu metode ML yang memiliki ciri khas dalam hal pembelajaran langsung pada lingkungan [16].

### 3. Algoritma Deep Q-Network

Deep Q-Learning merupakan salah satu basis algoritme RL dengan paradigma model free. Secara prinsip, algoritme ini mengasumsikan agent tidak mengetahui model matematis dari lingkungan yang akan dipelajari seperti peluang transisi probabilitas pada state tertentu dan reward apa yang akan didapatkan. Algoritme ini secara perlahan membangun ‘pengetahuan’ dari nol melalui proses interaksi dengan lingkungan [16].

### 4. Agile dan Scrum

Agile adalah metodologi pengembangan perangkat lunak yang mengutamakan iterasi, fleksibilitas, dan kolaborasi untuk merespon perubahan kebutuhan dengan cepat [17]. menyatakan bahwa Agile dipilih karena sifatnya yang adaptif dan responsif terhadap perubahan, dengan tingkat keberhasilan yang lebih tinggi dibandingkan metode tradisional seperti Waterfall [17]. Scrum merupakan kerangka kerja Agile yang memecah proses pengembangan menjadi siklus pendek disebut sprint (biasanya 1-4 minggu) [18]. menjelaskan bahwa dalam Scrum terdapat aktivitas harian seperti Daily Scrum dan pertemuan Sprint Review yang membantu tim berkolaborasi, mengevaluasi progres, dan menyesuaikan pekerjaan agar nilai hasil lebih optimal [18]. [19]menambahkan bahwa Scrum memberikan kerangka kerja terstruktur melalui

peran dan praktik tertentu yang mendukung penyelesaian masalah kompleks secara adaptif<sup>[19]</sup>

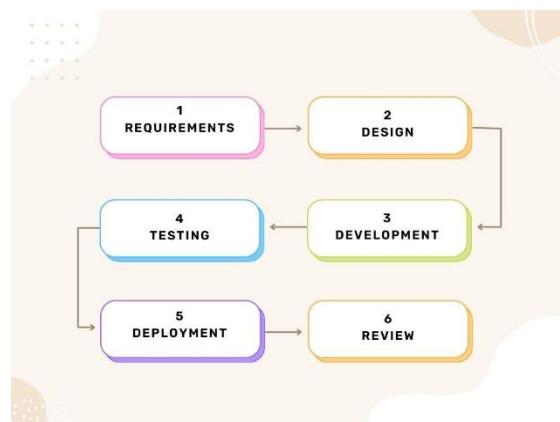
### 3. Metode Penelitian

#### 3.1. Metode Pengumpulan Data

Pada penelitian ini menggunakan teknik pengumpulan data dengan cara simulasi data sintesis. Data ini mencakup profil dinamis pengguna, data kebugaran dasar, dan simulasi *feedback* harian (*reward*). Pendekatan data sintesis dipilih untuk mengatasi isu privasi data sensitif dan memungkinkan simulasi skenario *feedback* yang kompleks dan konsisten, yang esensial untuk pelatihan algoritma *Reinforcement Learning*. Peneliti melakukan analisa kinerja DQN pada tahap pertama selama masa pelatihan yang melibatkan evaluasi *Reward Function* dan *Exploration Policy*. Tahap kedua adalah *Analisis Komparatif prototipe fungsional (Proof-of-Concept)* terhadap program kebugaran konvensional untuk mengukur tingkat optimalisasi program dan konsistensi pengguna di lingkungan simulasi.

#### 3.2. Metode Pengembangan sistem

Metode pengembangan yang digunakan oleh peneliti adalah **Agile Scrum**. Pendekatan ini dipilih karena mampu mengakomodasi sifat adaptif dari sistem rekomendasi berbasis Reinforcement Learning, memungkinkan perubahan kebutuhan, dan menghasilkan software *increment* secara berkala.



Gambar 1. Struktur Scrum

#### 1. Requirements

Di tahapan requirements, peneliti berfokus kepada penetapan kebutuhan sistem seperti; Menentukan profil user (level kebugaran, tujuan, kondisi medis, preferensi) sebagai **State** dalam model RL. Kemudian peneliti juga menetapkan tujuan utama yaitu membuat sistem yang harus mampu merekomendasikan suatu *action*(program latihan dan nutrisi) yang optimal guna mencapai *reward* (kemajuan kebugaran). Data yang dibutuhkan pada penelitian ini adalah data kebugaran dasar yang berisi berat, tinggi, bmr. Data *action* dan *reward* yang berupa *feedback* dari *user*.

Sistem rekomendasi adaptif kebugaran ini beroperasi berdasarkan siklus pembelajaran berulang Reinforcement Learning (RL), di mana empat kelas utama dalam diagram merepresentasikan alur kerja algoritma Deep Q-Network (DQN). Proses dimulai ketika agen memproses State S, yang mencakup profil dinamis pengguna seperti level kebugaran dan

data dasar tubuh. Berdasarkan *State* ini, agen memilih *Action A* berupa program latihan dan nutrisi yang optimal. Setelah *Action* dilakukan, sistem menerima *Reward R* yang merupakan umpan balik, mayoritas berupa nilai netral (0), yang kemudian dikirimkan ke modul *Update Q*. Pada tahap *Update Q* inilah kebijakan agen diperbarui—menggunakan atribut *nilaiQStabil* dan *lajuPenurunanEpsilon*—agar keputusan di siklus berikutnya menjadi lebih baik. Namun, hasil uji coba menunjukkan bahwa agen cenderung eksploratif, dengan dominasi tinggi pada Aksi 0, dan distribusi *Reward* yang sempit di sekitar nol, menandakan adanya kebutuhan mendesak untuk penyesuaian pada *lajuPenurunanEpsilon* dan implementasi *Reward Shaping* untuk mempercepat pembelajaran dan memastikan agen RL dapat mengambil keputusan yang lebih bervariasi.

## 2. Design

Di tahapan Design, peneliti mengubah kebutuhan sistem menjadi rancangan teknis dan rencana iterasi. Dimulai dari perancangan arsitektur sistem dengan menggunakan . Kemudian peneliti juga melakukan perancangan pada model reinforcement learning seperti : menetapkan *state*, menetapkan *action*, mendefinisikan *reward*, dan memilih algoritma (DQN)

## 3. Development

Di tahapan ini peneliti akan mengimplementasikan sistem secara teknis, Dimulai dari pemilihan bahasa pemrograman yang akan digunakan (python), merancang basis data guna menyimpan data pengguna, riwayat progres, dan reward yang di hasilkan. Peneliti membangun kerangka model RL dengan pustaka yang berasal dari pytorch juga melakukan training awal menggunakan simulasi

## 4. Testing

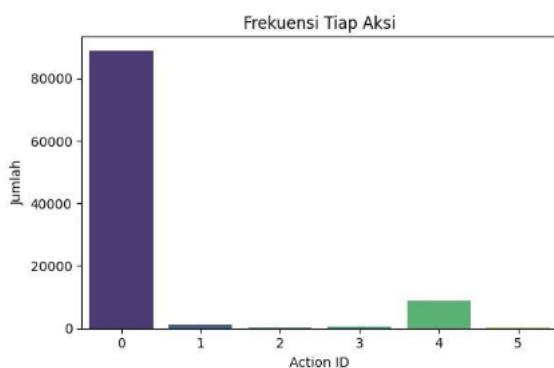
Di tahap ini peneliti melakukan uji simulasi guna memverifikasi konvergensi pada agen rl (apakah agen dapat belajar memilih action yang memberikan reward maksimum). Peneliti juga menganalisis *Reward Function* guna memastikan tidak ada perilaku *exploit* yang tidak diinginkan (misalnya, merekomendasikan program yang tidak aman demi *Reward* tinggi). Dan peneliti juga melakukan pengujian fungsional dan kinerja pada sistem guna memastikan semua fitur berjalan sesuai dengan kebutuhan, juga guna mengukur waktu respons agen rl agar optimal saat digunakan real-time.

## 5. Deployment dan Review

Pada tahap ini peneliti akan menjelaskan lingkungan penerapan akhir, dan juga penulis akan menerapkan mekanisme feedback loop guna memastikan agen agen rl terus belajar dan beradaptasi

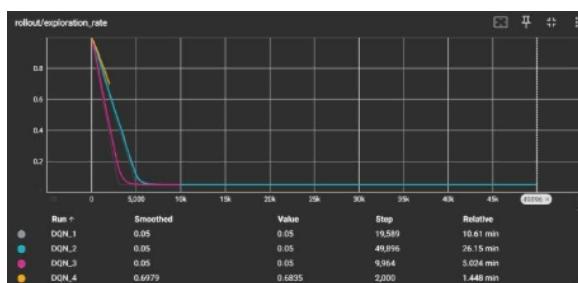
## 4. Hasil dan Pembahasan

### 4.1. Hasil Penelitian



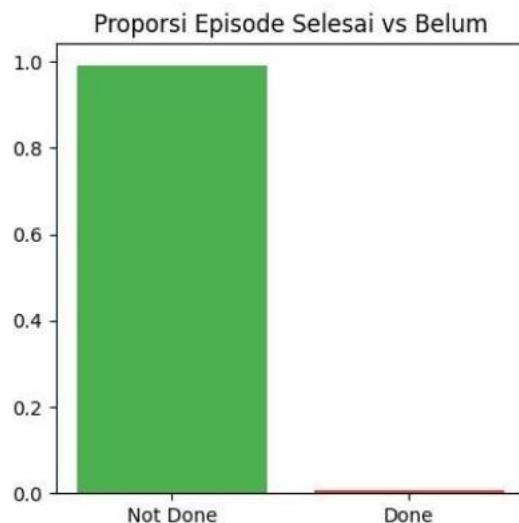
Gambar 5. Grafik Histogram Frekuensi Tiap Aksi

Berdasarkan uji coba yang ditunjukkan oleh diagram histogram yang Menunjukkan seberapa sering setiap aksi (0–5) diambil oleh agen selama training. Aksi 0 mendominasi jauh lebih banyak (>85%). Yang berarti, agen sangat sering memilih aksi yang sama — bisa jadi itu aksi “netral” atau “aman”. Aksi lain seperti 4 terlihat lumayan sering, sedangkan aksi 1–3–5 jarang muncul. Kesimpulan: Agen cenderung belum seimbang dalam eksplorasi — bisa jadi karena reward aksi 0 paling stabil, atau karena epsilon (eksplorasi) sudah terlalu rendah di akhir training.



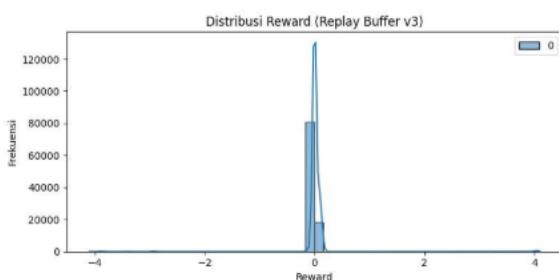
Gambar 6. Epsilon Eksplorasi

Seperti yang ditunjukkan pada grafik eksplorasi pada gambar 6 yang menunjukkan bahwa diakhir training memang epsilon berada dititik terendah.



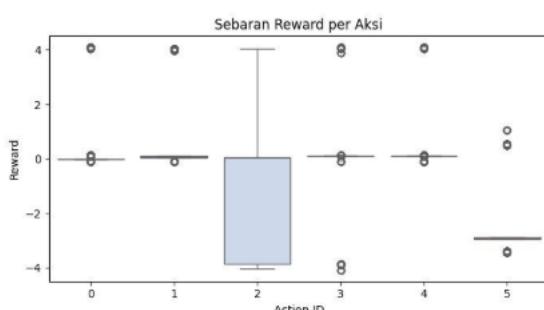
Gambar 7. Grafik Histogram Proporsi Episode

Berdasarkan hasil grafik histogram pada gambar 7 Menunjukkan proporsi episode yang berakhir (done=True) dibandingkan yang masih berjalan. Hampir semua data adalah Not Done, artinya sebagian besar interaksi masih di tengah episode. Hanya sedikit transisi yang menandai akhir episode (done=True). Replay buffer lebih banyak berisi pengalaman jangka menengah, bukan terminal states. Ini bagus untuk stabilitas DQN, tapi kamu bisa menambah variasi episode pendek agar agen belajar konteks akhir percakapan juga.



Gambar 8. Grafik Histogram Distribusi Reward

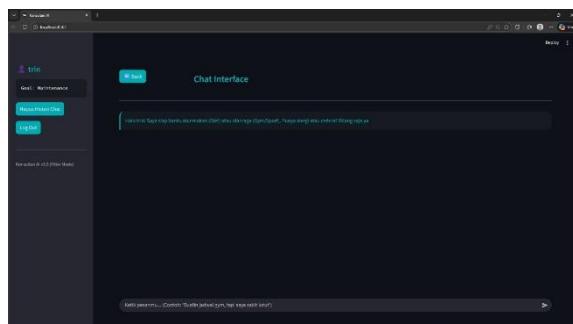
Berdasarkan hasil pada grafik histogram pada gambar 8 Menunjukkan bagaimana reward tersebar di seluruh replay buffer. Gambar tersebut menggambarkan Mayoritas reward berkisar di sekitar 0 (netral) dengan sedikit outlier di sekitar +4 dan -4. Distribusi sangat sempit di tengah, menunjukkan reward jarang ekstrem. Agen lebih sering menerima feedback kecil, artinya environment cukup stabil. Namun, untuk mempercepat pembelajaran, kamu bisa pertimbangkan reward shaping (memperjelas perbedaan antara aksi baik/buruk).



Gambar 9. Grafik Box Plot Sebaran Reward

Berdasarkan grafik box plot memetakan variasi reward yang diterima untuk setiap aksi (boxplot). Aksi 0, 1, 3, dan 4 cenderung memiliki reward stabil di sekitar 0. Aksi 2 menunjukkan variasi besar (reward bisa sangat positif atau negatif).

Aksi 5 sering memberi reward negatif. Ada aksi yang berisiko tinggi tapi berpotensi tinggi (aksi 2) — mungkin perlu strategi eksplorasi khusus. Aksi 5 tampak “buruk”, bisa jadi perlu di-review apakah benar fungsinya memberi kontribusi yang valid dalam environment.



Gambar 10. Frontend Program

Berdasarkan gambar 10 user dapat melakukan interaksi bersama agen DQN. Seperti user melakukan input state dan beberapa kondisi lainnya dan kemudian meminta informasi dari agen DQN. Agent DQN akan menerima input dari user dan akan mencari informasi berdasarkan inputan tersebut, dan kemudian agen DQN akan memberikan informasinya melalui chat box yang tersedia.

## 4.2 Pembahasan

Peneliti berhasil merancang dan mengembangkan prototipe fungsional Sistem Rekomendasi Latihan dan Nutrisi Adaptif menggunakan metodologi Design Science Research Methodology (DSRM), dengan Agile Scrum sebagai kerangka kerja pengembangan.

Algoritma inti yang diimplementasikan adalah Deep Q-Network (DQN), dipilih karena kemampuannya dalam pembelajaran adaptif dan berkelanjutan, serta mendukung *sequential decision-making*. Agen DQN beroperasi berdasarkan *State* pengguna, yang mencakup profil dinamis seperti tingkat kebugaran, tujuan, kondisi medis, serta data dasar tubuh (berat, tinggi, BMR). Berdasarkan *State* ini, agen memilih *Action* berupa rekomendasi program latihan dan nutrisi yang optimal. Seluruh sistem dilatih dan dievaluasi menggunakan data sintetis untuk mengatasi isu privasi data sensitif sambil memungkinkan simulasi skenario *feedback* yang kompleks. Namun, analisis kinerja awal pada tahap pengujian simulasi menunjukkan tantangan signifikan dalam proses pembelajaran agen.

## 5. Kesimpulan

Penelitian ini bertujuan mengatasi keterbatasan program kebugaran konvensional yang bersifat generik dan statis, melalui perancangan dan pengembangan Sistem Rekomendasi Latihan dan Nutrisi Adaptif yang dipersonalisasi menggunakan pendekatan Reinforcement Learning (RL), khususnya algoritma Deep Q-Network (DQN). Arsitektur sistem cerdas ini mampu mengelola data dinamis pengguna untuk menghasilkan rekomendasi program yang optimal secara berkelanjutan. Prototipe fungsional (*Proof-of-Concept*) sistem telah diuji menggunakan data

sintetis untuk mensimulasikan skenario *feedback* yang kompleks. Namun, hasil uji coba menunjukkan bahwa agen cenderung belum seimbang dalam eksplorasi, terlihat dari dominasi yang tinggi pada Aksi 0 (>85%). Selain itu, mayoritas *reward* yang diterima agen berkisar di sekitar nilai netral (0) dan hanya sedikit interaksi yang mencapai *terminal states*. Keterbatasan ini mengindikasikan perlunya penyesuaian lebih lanjut pada parameter eksplorasi dan definisi *reward function* (*reward shaping*) untuk mempercepat pembelajaran dan memastikan agen RL dapat mengambil keputusan yang lebih bervariasi dan optimal.

Oleh karena itu, untuk meningkatkan kinerja dan adaptabilitas sistem berbasis DQN ini, peneliti disarankan untuk berfokus pada tiga aspek utama. Pertama, perlu dilakukan penyesuaian pada kebijakan eksplorasi (*exploration policy*). Mengingat dominasi tinggi Aksi 0 dan rendahnya nilai *epsilon* di akhir *training*, disarankan untuk meningkatkan atau menyesuaikan laju penurunan *epsilon* agar agen memiliki lebih banyak kesempatan untuk mencoba Aksi yang jarang muncul (Aksi 1, 3, 5). Kedua, perlu dipertimbangkan implementasi Reward Shaping. Karena mayoritas *reward* berkisar di sekitar nilai netral (0), memperjelas perbedaan antara *reward* positif dan negatif yang signifikan akan mempercepat proses pembelajaran dan mengurangi kecenderungan agen untuk memilih

aksi "aman". Ketiga, perlu dilakukan peninjauan dan redefinisi pada Aksi yang menunjukkan perilaku tidak diinginkan, seperti Aksi 5 yang sering memberikan *reward* negatif, dan merancang strategi eksplorasi khusus untuk Aksi 2 yang berisiko tinggi tetapi berpotensi *reward* tinggi, sehingga agen dapat memanfaatkan potensi optimalisasi tanpa mengorbankan keselamatan pengguna.

penyesuaian lebih lanjut pada parameter eksplorasi dan definisi *reward function (reward shaping)* untuk mempercepat pembelajaran dan memastikan agen RL dapat mengambil keputusan yang lebih bervariasi dan optimal

## 6. Daftar Pustaka

- [1] Sobri muhammad et al, "Systematic literature review untuk membuat model aplikasi pemantauan kesehatan cardiovascular," vol. 2, no. 2, pp. 458–464, 2018.
- [2] R. A. Putra, "Analisis deep q-network (DQN) untuk simulasi lampu lalu lintas adaptif berdasarkan waktu tunggu kendaraan," 2024.
- [3] Waisen et al., "Perancangan sistem rekomendasi kurikulum personal berbasis CLT dan RL untuk edukasi daring," vol. 14, pp. 1189–1198, 2025.
- [4] M. Rizqi and A. Zayyad, "Penerapan metode deep learning pada sistem rekomendasi film".
- [5] A. D. Hartono and J. Sutopo, "Implementasi algoritma upper confidence bound untuk sistem rekomendasi musik," pp. 82–90, 2024, doi: 10.33364/algoritma/v.21-2.2095.
- [6] Kurniawan Denni et.al, "Sistem rekomendasi keterampilan dengan metode user-based collaborative filtering dan tanimoto coefficient similarity sesuai . sehingga banyak para pencari kerja yang tidak update terhadap jenis menggunakan teknik user based collaborative filtering dan tan," vol. XVI, pp. 52–62, 2021.
- [7] Sartika Devi et. al, "Analisis kinerja sistem rekomendasi yang menggunakan collaborative filtering berdasarkan pengguna dengan python," vol. 3, no. 1, pp. 4686–4696, 2025.
- [8] Wijayanto Ardhi et. al, "Recomendation system with content-based filtering method for culinary tourism in," vol. 8, no. 2, pp. 65–72, 2019.
- [9] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions," vol. 17, no. 6, pp. 734–749, 2005.
- [10] S. V. K. et Al., "Sistem rekomendasi resep masakan menggunakan metode content based filtering berdasarkan preferensi pengguna," vol. 8, no. 2, pp. 345–355, 2025.
- [11] Komarudin Alim et.al, "Jurnal olahraga indragiri," vol. 10, no. 08, pp. 97–109, 2023.
- [12] M. Huang and R. T. Rust, "Artificial intelligence in service," vol. 21, no. 2, pp. 155–172, 2018, doi: 10.1177/1094670517752459.
- [13] A. S. Pratikno, "Implementasi artificial intelligence dalam memetakan karakteristik , kompetensi , dan perkembangan psikologi siswa sekolah dasar melalui platform offline implementasi artificial intelligence dalam memetakan karakteristik , kompetensi , dan perkembangan ps," no. September 2017, 2018.
- [14] A. Raup, W. Ridwan, Y. Khoeriyah, and Q. Y. Zaqiah, "Deep learning dan penerapannya dalam pembelajaran," vol. 5, no. September, pp. 3258–3267, 2022.
- [15] J. Andreanus and A. Kurniawan, "Sejarah , teori dasar dan penerapan reinforcement learning : Sebuah tinjauan pustaka," vol. 12, no. 2, pp. 113–118, 2020.

- [16] R. Febrian, E. Setiawan, and B. H. Prasetio, "Simulasi metode dueling double deep q-learning pada unmanned aerial vehicle untuk menghindari halangan," vol. 6, no. 3, pp. 1499–1506, 2022.
- [17] D. Murdiani, A. Yudhana, T. Informatika, and U. A. Dahlan, "Implementasi agile method dalam pengembangan jurnal elektronik di lembaga penelitian non pemerintahan (NGO) implementation of agile method in development of electronic journal in non government research institution (NGO)," vol. 7, no. 4, pp. 709–718, 2020, doi: 10.25126/jtiik.202071839.
- [18] H. Santoso, D. Pungki, A. Aziz, and A. Zaini, "Implementasi agile scrum pada proses pengembangan aplikasi monitoring MBKM di UNIKAMA," vol. 4, no. 4, pp. 208–215, 2022.
- [19] D. A. Wasesha, "Implementasi scrum framework pada informasi pemesanan jasa inspeksi mobil bekas," vol. 7, no. 3, pp. 842–853, 2023, doi: 10.52362/jisamar.v7i3.903.